The PHENIX Event Builder

David Winter

Columbia University

for the PHENIX Collaboration

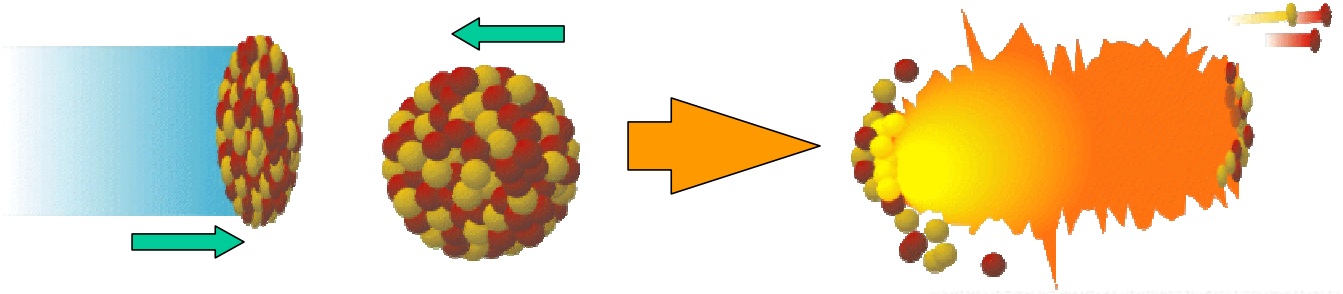
DNP 2004

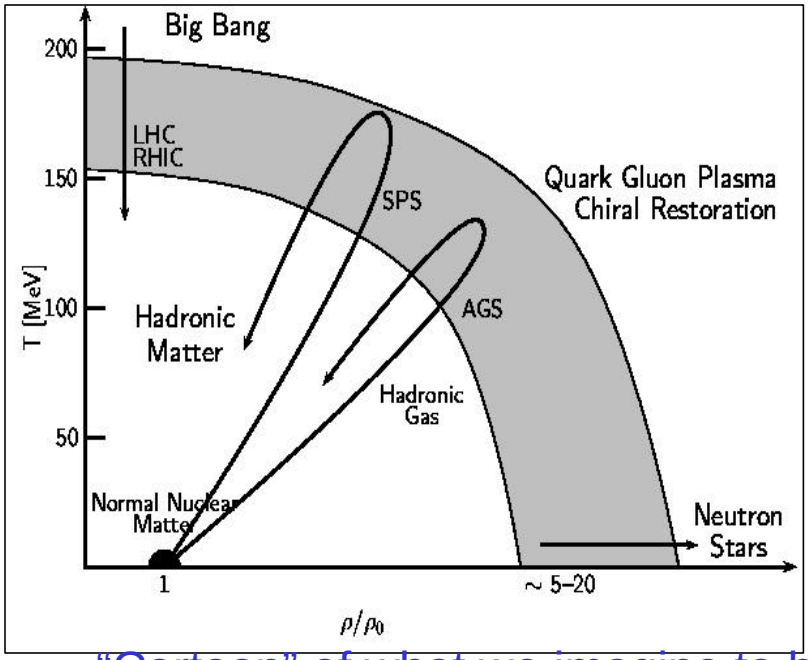
Chicago, IL

Overview

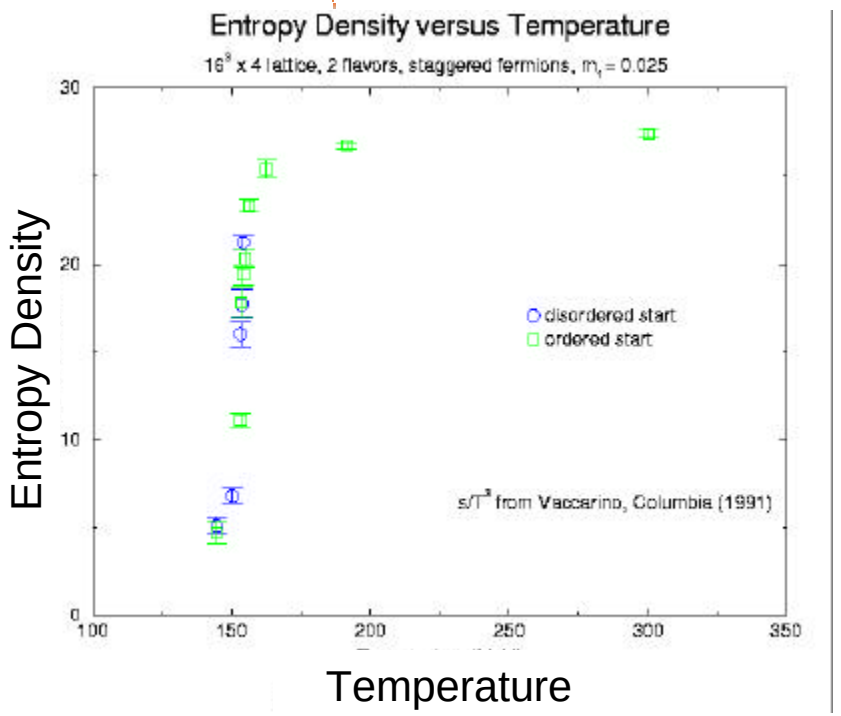
- Introduction
  - Quarks & Gluons at the extreme: Heavy Ion Collisions at RHIC
  - The challenge: The PHENIX experiment and its DAQ
- The Event Builder
  - Software & Hardware
  - System Design
  - Monitoring & Performance
- Present and Future Development
- Summary

Torturing the Nucleus: Heavy Ion Collisions





"Cartoon" of what we imagine to be phase diagram of hadronic matter (Temp vs. baryon density)



Lattice QCD calculations have long indicated existence of phase transition

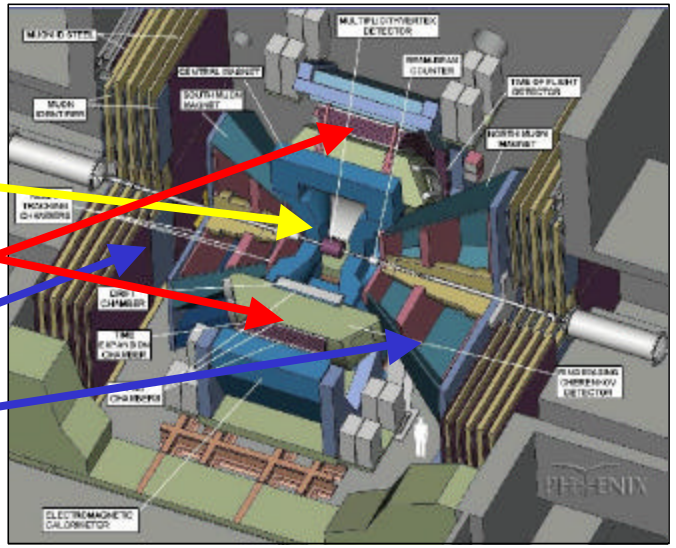
PHENIX @ RHIC

- Two independent rings
- 3.83 km circumference
- Capable of colliding ~ any nuclear species on ~ any other species
- Center of Mass Energy:
  - ➔ 500 GeV for p-p
  - ➔ 200 GeV for Au-Au (per N-N collision)
- Luminosity
  - Au-Au: 2 x 10^{26} cm^{-2} s^{-1}
  - p-p : 2 x 10^{32} cm^{-2} s^{-1} (polarized)

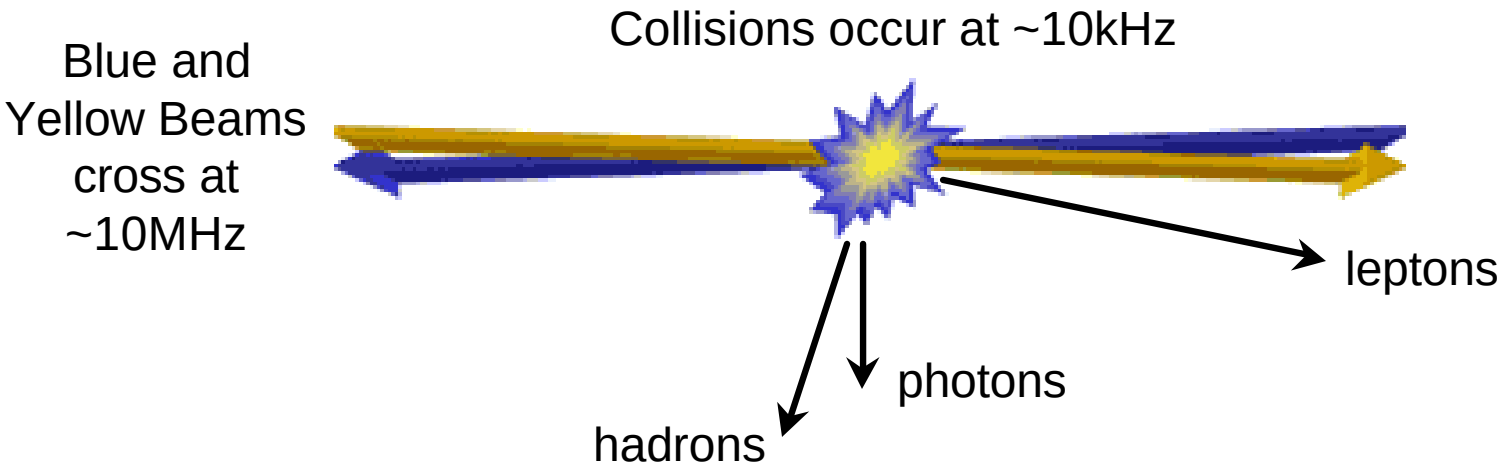
Event characterization detectors in center

Two central arms for measuring hadrons, photons and electrons

Two forward arms for measuring muons



Data Collection: The Challenge



- High rates
- Large event sizes (Run-4: >200 kb/event)
- Interest in rare physics processes

  => ***Big Headache***
- How do we address these challenges?
  - Level-1 triggering
  - Buffering & pipelining: “deadtime-less” DAQ
  - High Bandwidth (Run-4: ~400 MB/s archiving)
  - Fast processing (eg. Level-2 triggering)

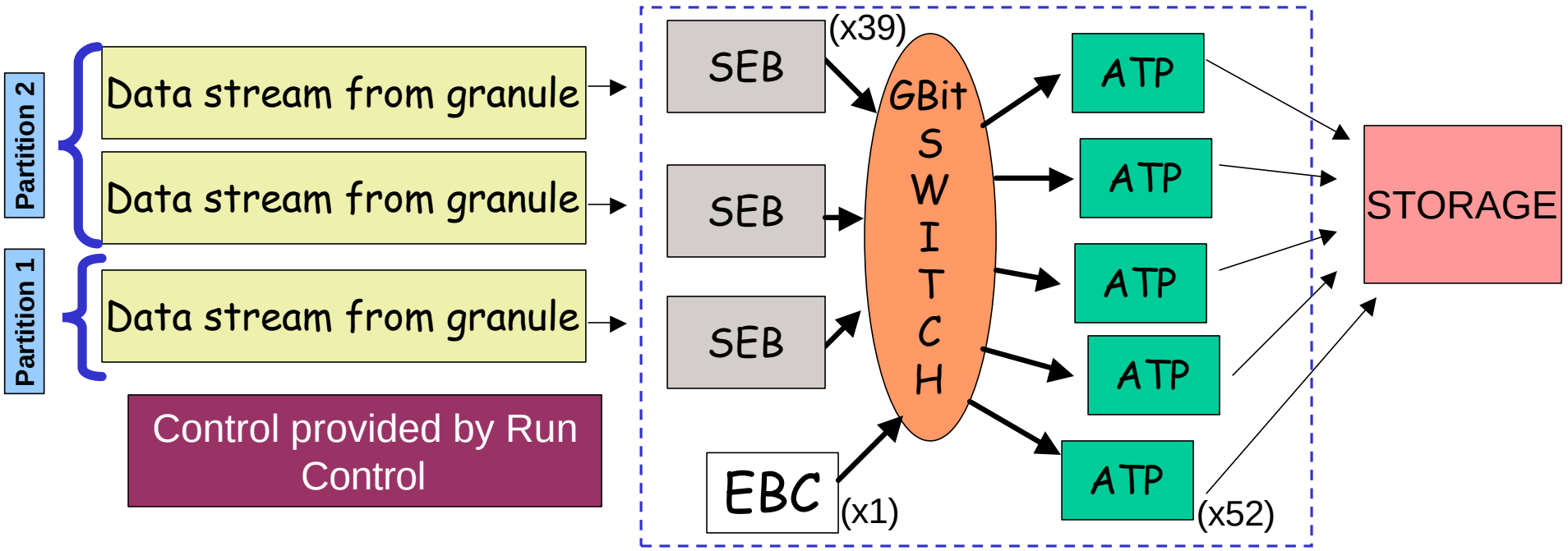
Run-4

1.5 Billion Events
300-400 MB/s
~200 kB/event
2-2.5 kHz rate

Run-5

~200 kB/event
5 kHz rate
⇨1 GB/s !!

Event Builder Overview

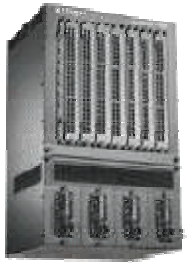


- Three functionally distinct programs derived from the same basic object
- SubEvent Buffer (SEB): Collects data for a single subsystem – a “subevent”
- Event Builder Controller (EBC): Receives event notification, assigns events, flushes system
- Assembly Trigger Processor (ATP): Assembles events by requesting data from each SEB, writes assembled events to short-term storage, can also provide Level-2 trigger environment

Software & Hardware

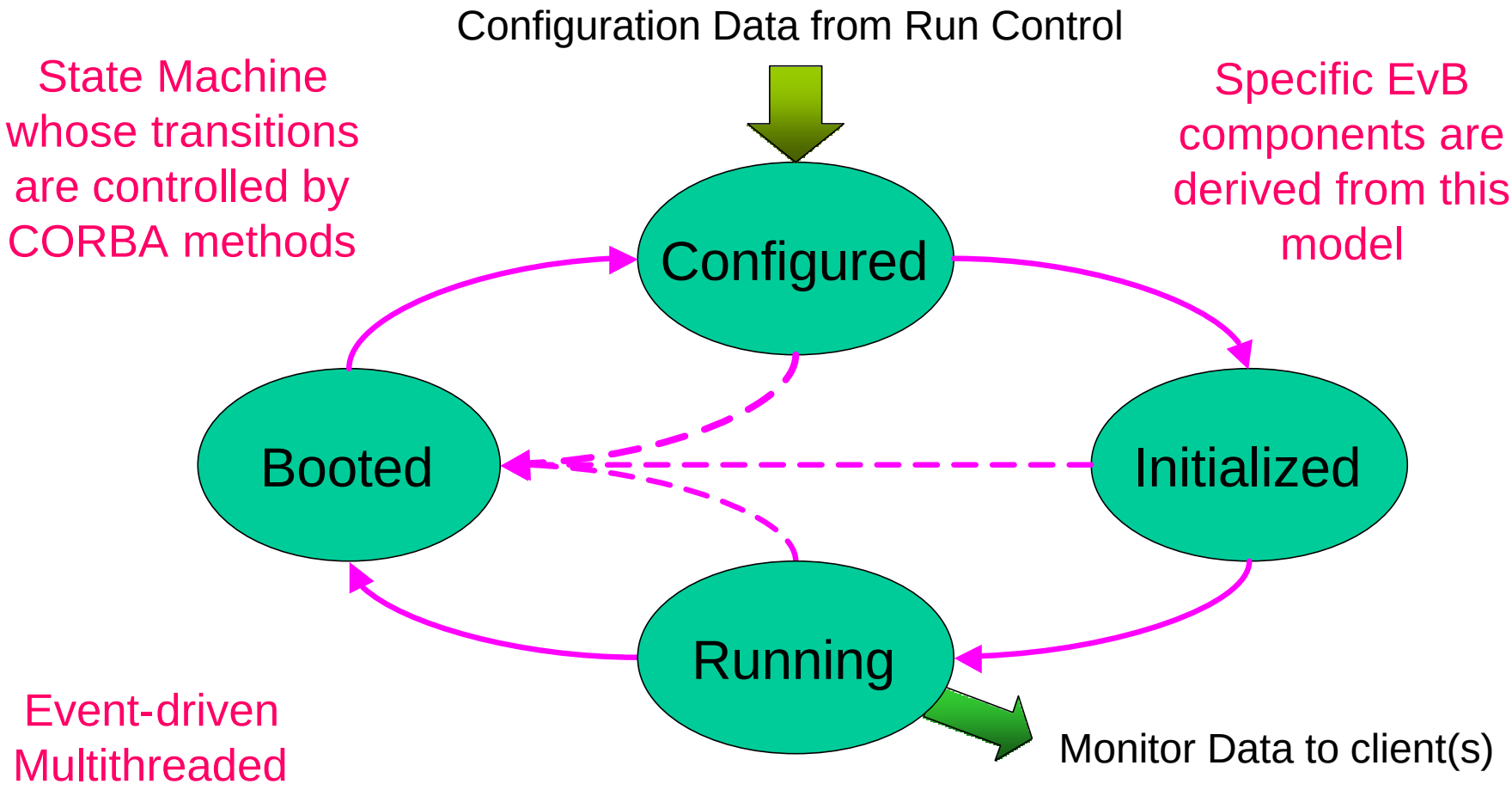
- Software environment: Run-4 to Run-5 paradigm shift
  - New platform: Windows NT/2k ➡Linux 2.4.x (FNAL's SL3.0.2)
  - New compiler: Visual C++ 6.0 ➡GCC 3.2.3
  - Same: Iona Orbix (CORBA), Boost template library

- 105 1U Rack-mounted dual CPU x86 servers
  - 1.0 GHz PIII & 2.4 GHz P4 Xeon
  - Gigabit NIC (Intel PRO/1000 MT Server)
- Foundry FastIron 1500 Gigabit Switch
  - 480 Gbps total switching capacity
  - 15 Slots, 10 in use (includes 96 Gigabit ports)



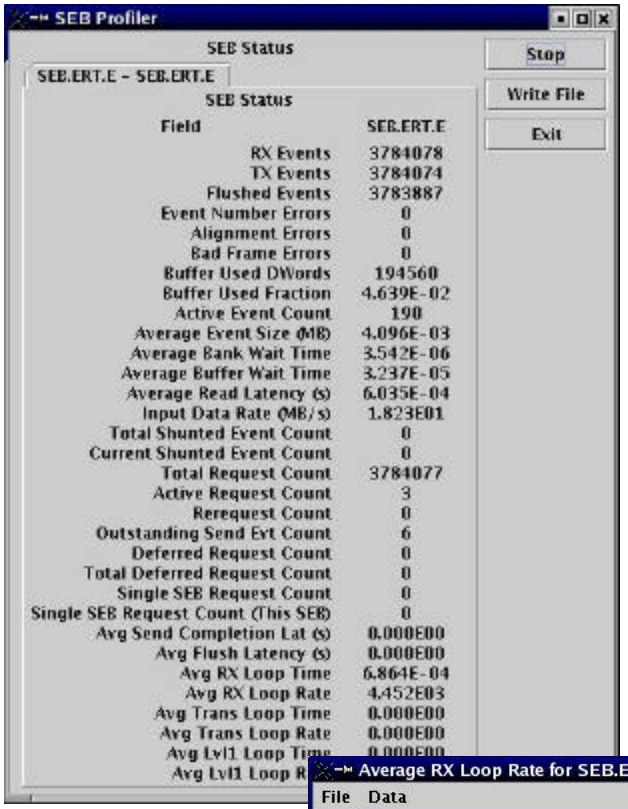
- JSEB: custom-designed PCI card
  - Interface between EvB and incoming data stream
  - Dual 1 MB memory banks (allows simultaneous r/w)
  - Programmable FPGA
  - Latest firmware enables DMA Burst – up to 100 MB/s I/O

Basic Component Design

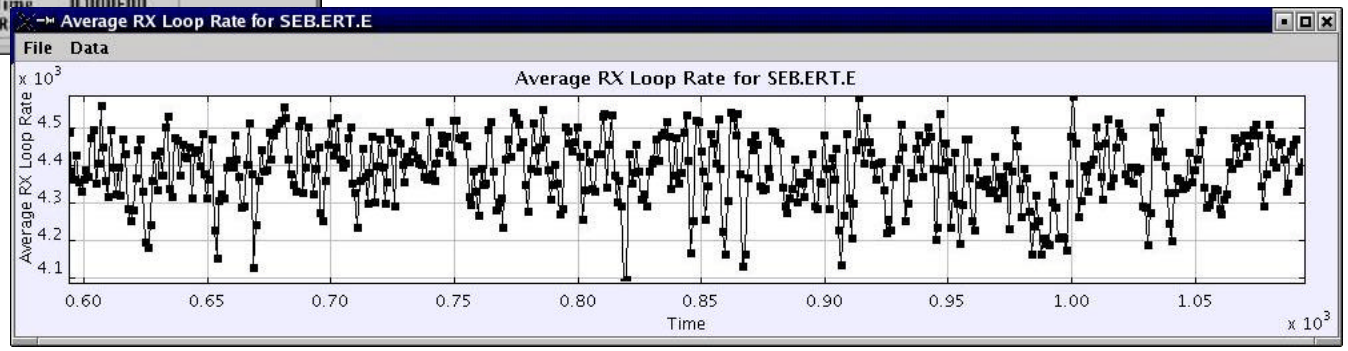


UDP sockets for data TCP sockets for control

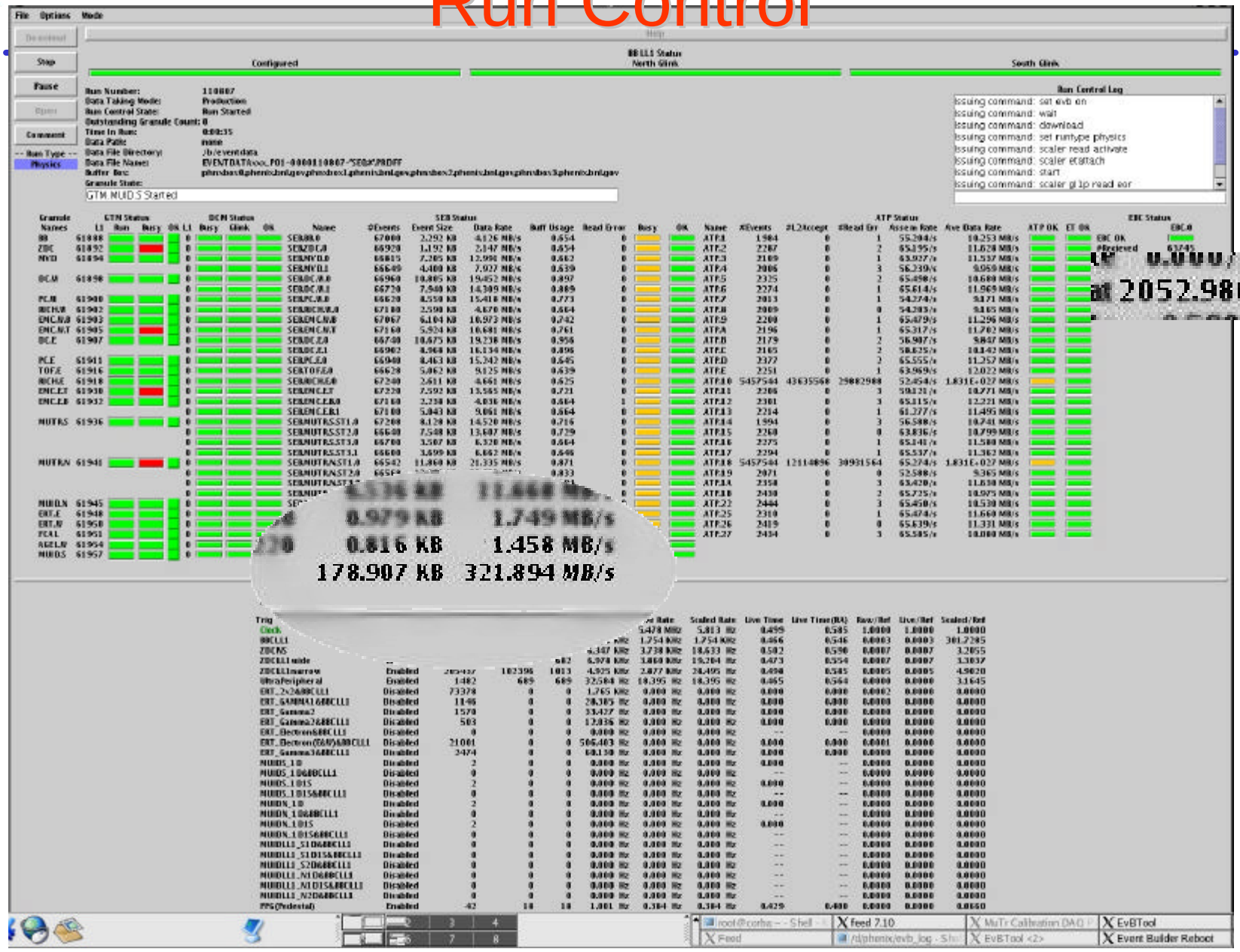
Performance Monitoring



- Each component keeps track of various statistics
- Data served via CORBA calls
- Java client displays stats in “real time”
- Strip charts display data as function of time
- Histograms display data as function of component



Run Control


Configured
North Glink
South Glink
Run Control Log
GTM MUIDS Started
0.979 KB 1.749 MB/s
0.816 KB 1.458 MB/s
178.907 KB 321.894 MB/s
at 2052.980

Where does the future lie?

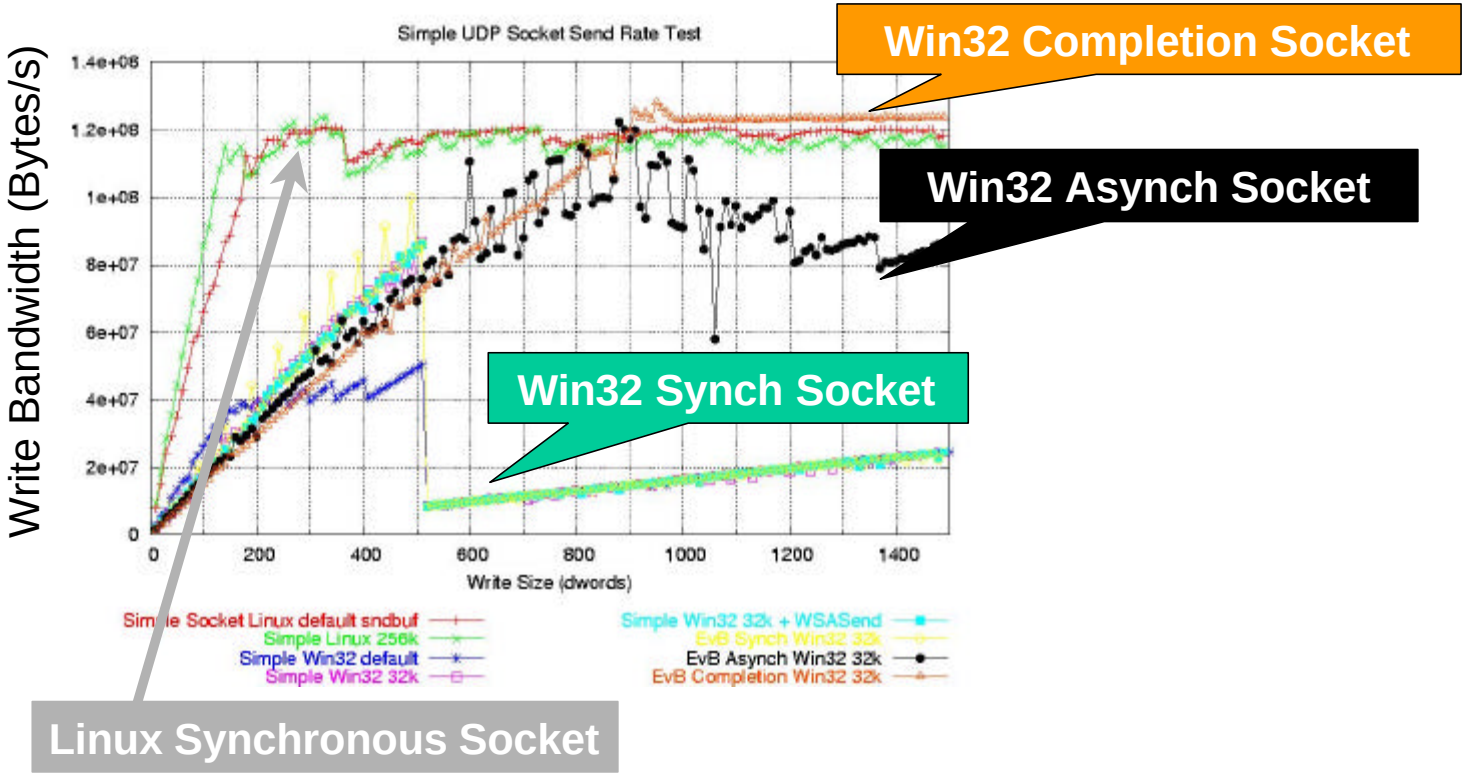
How do we break the 2.5 kHz boundary?

The most important improvement we can make: Port to Linux

- Win32 ***was*** the right platform when using ATM
  - ATM (completely) replaced by Gigabit in Run-4
- At the limit of what Win32 can provide us
- Growing pains while porting
  - Thread-safety: Replacing Interlocked operations
    - Who said writing atomic operations in assembly isn't fun?
  - Replacing Overlapped socket I/O with synchronous I/O
    - Linux and AIO? Maybe in our lifetime...
  - Event synchronization: Events vs. Condition variables
  - Timeout mechanisms (eg. Dropped packets/events)

The Impact of a Linux Port

A picture is worth a thousand words



Linux beats Win32 hands-down in simple socket tests

The Payoff

- Tests with multiple granule partitions have been performed
- Fake Data Mode with 1 & 2 granules
  - 26 kHz (0.240 kB/event) to 10 kHz (~150 kB/event)
- Clock Triggers with 1, 2, & 4 granule partitions
  - 4.5 - 5 kHz (little to no dependence on event size)

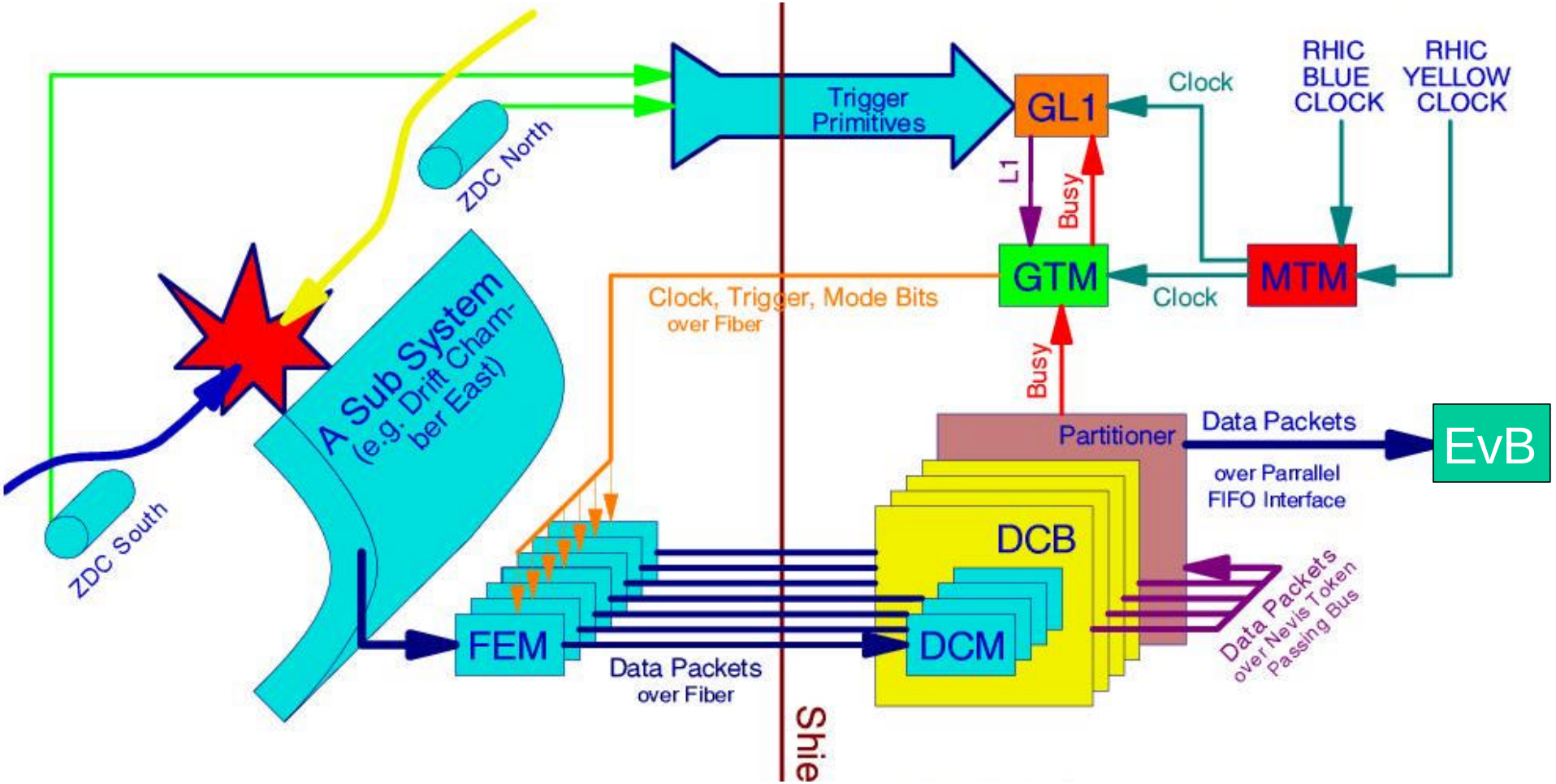
Summary

- Ideal laboratory for the study of hot, dense quark matter: Heavy Ion Collisions at RHIC.
- The PHENIX experiment is designed to make high statistics measurements of a variety of physics processes, esp. rare signatures
- The PHENIX Event Builder lies at the heart of a parallel pipelined DAQ, enabling high rates of archiving.
  - Three multithreaded programs originally implemented on Win32
  - Win32 EvB has done a respectable job so far, but we need more
- Linux is the future of the PHENIX EvB
  - Synchronous I/O superior to even Win32's overlapped I/O
  - OS overheads much lower (in general)
  - Various issues when porting from Win32 to Linux
    - I/O, timers, threading
- Bottom line: Run-5 will have a Linux Event Builder that early tests show will improve performance by as much as a factor of 10. The goal of archiving up to 1 GB/s at 5 kHz is well within reach.

Backup Slides

The PHENIX DAQ

A **granule** is a detector subsystem, including readout electronics



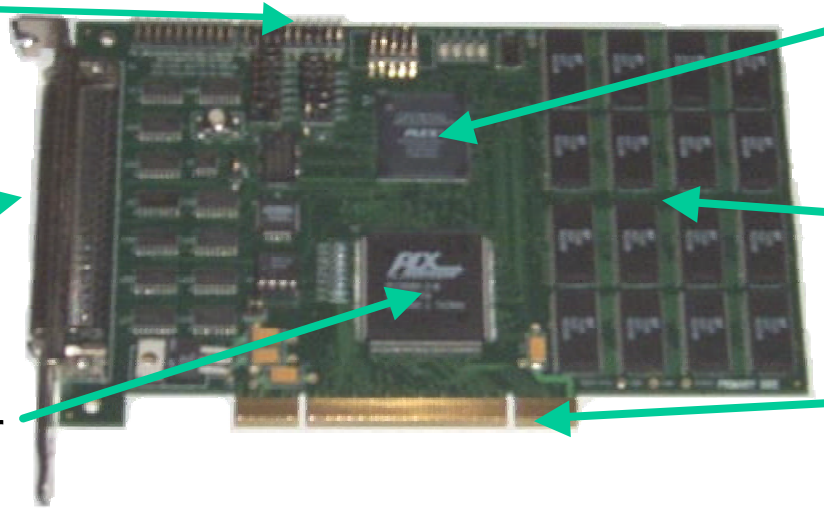
A **partition** is one or more granules that receives the same triggers & busies

JSEB Interface Card

JTAG port for programming FPGA

Input for JSEB cable from DAQ

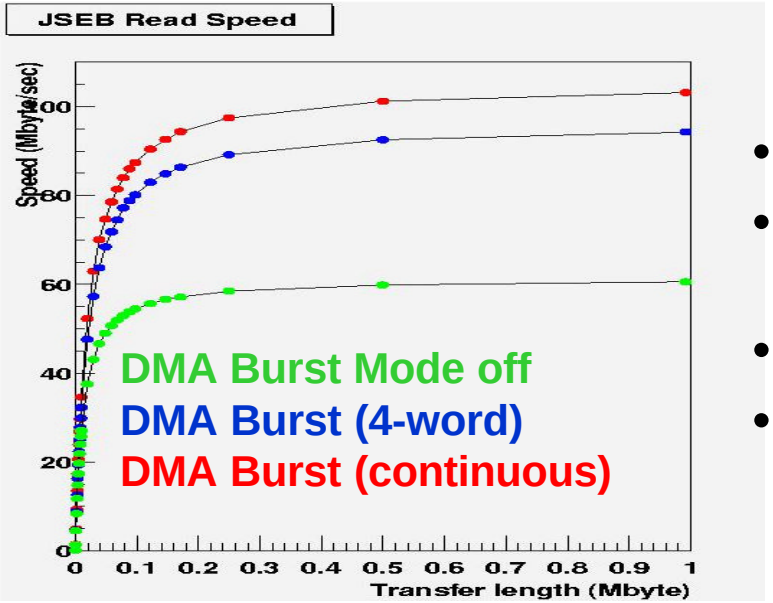
PLX 9080 PCI controller



Altera FLEX10k FPGA

2 banks of 1 MB static RAM

PCI 2.1 interface: 32-bit 33 MHz with 3.3v or 5v signaling



- Interfaces the DAQ to the SEB
- Data transmitted via 50-pair 32-bit parallel cable
- (Pseudo) Driver provided by Jungo
- Latest firmware provides DMA burst mode

CORBA

Common Object Request Broker Architecture

- The networking protocol by which the run control software components talk to each other.
- Based on a client/server architecture through a heterogeneous computing environment.
  - VxWorks, Linux, Windows NT/2000
- Servers: implement “CORBA Objects” that execute the functionality described in the member functions of the object
- Clients: invoke local CORBA Object’s methods. This causes the server to execute the code of its corresponding member function

Characterizing PCI bus interactions

JSEB contention read test
Without network writing (top curves)
With network writing (bottom curves)